**Министерство образования и науки Российской Федерации**

**федеральное государственное автономное образовательное учреждение высшего образования**

**“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”**

**ФАКУЛЬТЕТ СРЕДНЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ**

**ОТЧЕТ**

**по Лабораторной работе**

**«Программная реализация математического метода оптимизации»**

Специальность 09.02.07 «Информационные системы и программирование»

Дисциплина «Математические методы»

|  |  |
| --- | --- |
| Преподаватель:  Коцюба И. Ю.  «30» ноября 2019г.  Оценка: | Выполнил:  студент группы Y2435  Котлицкий С.А. |

Санкт-Петербург

2019/2020

Задание

Реализовать программную реализацию поиска метрических характеристик программ.

Выполнение

В ходе выполнения лабораторной работы были выполнены следующие действия:

Была проанализирована теория, найдены основные формулы:

1. Объем программы ;
2. Минимально возможный объем программы ;
3. Уровень реализации программы L =.

Была создана программа с веб интерфейсом для нахождения метрических характеристик программы, написанной на языке JavaScript.

Для демонстрации работы программы рассмотрим два примера.

Проанализируем с помощью программы следующий код:

let func = (value) => {

let time = new Date().getHours();

if (time < 8 || time >= 22) {

this.sleep = true;

this.render();

} else {

this.sleep = false;

}

console.log('check\_sleep', this.sleep);

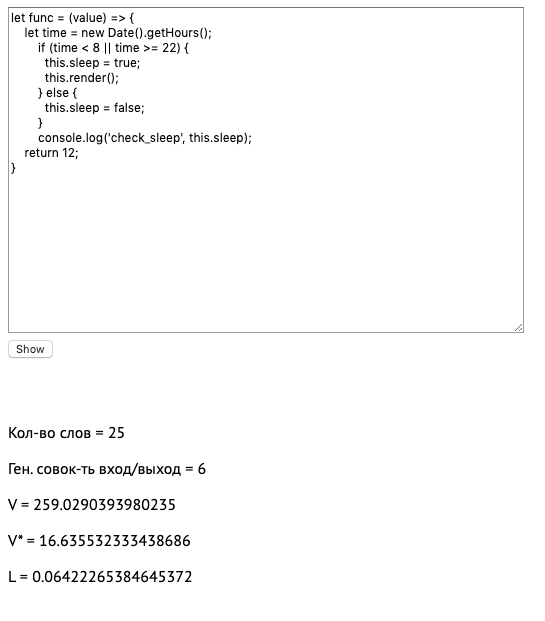
return 12;

}

Результат работы программы представлен ниже (Рисунок 1).

Из результатов можно сделать вывод, что:

* программа состоит из 25 слов;
* генеральная совокупность входных и выходных переменных равна шести;
* объем программы равен 259 единицам;
* минимально возможный объем программы равен 16.6 единицам;
* уровень реализации программы равен 0.064.



*Рисунок 1 Результат работы программы*

Проанализируем с помощью программы следующий код:

let func = (value) => {

const canvas = document.querySelector('#canvas');

const ctx = canvas.getContext('2d');

ctx.clearRect(0, 0, canvas.width, canvas.height);

image = this.sick ? sickImage : this.sleep ? sleepImage : activeImage;

ctx.drawImage(image, this.x, this.y, this.imageSize, this.imageSize);

this.poops.map(poop => {

ctx.drawImage(poopImage, poop.x, poop.y, 14, 14);

});

isHungerButton.disabled = this.sleep;

let time = new Date().getHours();

if (time < 8 || time >= 22) {

this.sleep = true;

this.render();

} else {

this.sleep = false;

}

console.log('check\_sleep', this.sleep);

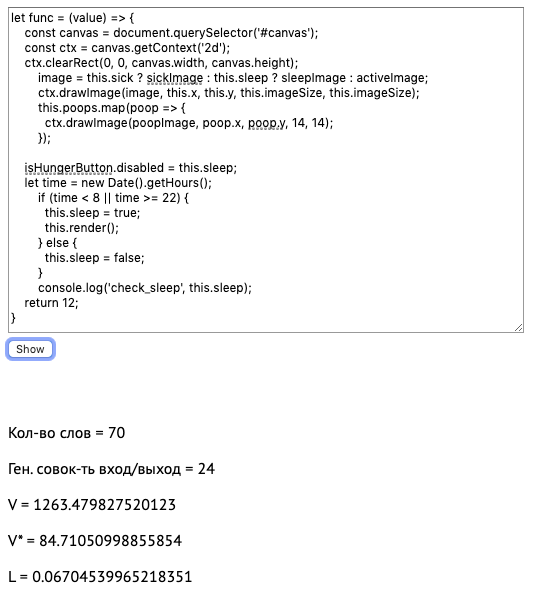
return 12;

}

Результат работы программы представлен ниже (Рисунок 2).

Из результатов можно сделать вывод, что:

* программа состоит из 70 слов;
* генеральная совокупность входных и выходных переменных равна 24;
* объем программы равен 1263.47 единицам;
* минимально возможный объем программы равен 84.7 единицам;
* уровень реализации программы равен 0.067.



*Рисунок 2 Результат работы программы*

Следует заметить, что из полученных результатов можно сделать вывод, что уровень реализации программы сильно отличается от единицы в сторону нуля, из чего можно сделать вывод, что формула для оценки уровня реализации программы не объективно оценивает код программы, написанной на языке JavaScript.

Листинг программы, файл index.js:

// Получаем доступ к html элементам

let words\_count = document.querySelector('#words\_count');

let in\_out = document.querySelector('#in\_out');

let v = document.querySelector('#v');

let v\_ = document.querySelector('#v\_');

let l = document.querySelector('#l');

// Инициализация переменных

let data = '';

let filtered = 0;

let input = 0;

const setData = value => {

data = value;

};

const setFiltered = value => {

filtered = value;

};

const setInput = value => {

input = value;

};

const hanldeChange = e => {

// console.log(e.length);

setData(e);

};

const form = document.getElementById('form');

form.addEventListener('submit', e => {

e.preventDefault();

setFiltered(

data

// Убираем из текста все спец. символы (кол-во слов в тексте)

.replace(/\.|\{|\}|\(|\)|\[|\]|!|\&|;|,|<|>|\?|\\*|\:|\'/gi, ' ')

.split(' ')

// Убираем из текста пробелы

.filter(el => el.trim().length > 2).length);

setInput(

// Поиск в тексте скобок с входными значениями в функции (входные значния)

(data.replace(/\s/g, '').match(/\((.\*?)\)/gi) || [])

// Убираем из текста скобки

.map(el => el.replace(/\)|\(/gi, ''))

.filter(el => el.length)

// Вычисление кол-ва входных значений

.reduce((acc, el) => {

if (el.split(',')) {

el.split(',').map(item => acc++);

} else {

acc++;

}

return acc;

}, 0) +

// Поиск входных данных в стрелочных функциях (входные значения)

(data.match(/\((.\*?)\=>/gi) || [])

// Убираем из текста спец. символы

.map(el => el.replace(/\)|\(|>|\=/gi, '').trim())

.filter(el => el.length).length +

// Поиск ключевого слова для возвращения значений (выходные данные)

(data.replace(/\s/g, '').match(/return(.\*?);/gi) || []).length +

// Проверка текста программы на наличие стрелочных функций в укороченной записи (выходные данные)

(data.replace(/\s/g, '').match(/\=>(.\*?)\)/gi) || []).filter(el => !el.match(/\=>{/)).length +

// Фильтруем текст программы на соответсвие перебирающих методов массивов (выходные данные)

(data.replace(/\s/g, '').match(/\.map|\.forEach|\.reduce/gi) || []).length

);

words\_count.innerHTML = `Кол-во слов = ${filtered}`;

in\_out.innerHTML = `Ген. совок-ть вход/выход = ${input}`;

v.innerHTML = `V = ${filtered \* Math.log(filtered) \*\* 2}`;

v\_.innerHTML = `V\* = ${(input + 2) \* Math.log(input + 2)}`;

l.innerHTML = `L = ${((input + 2) \* Math.log(input + 2)) / (filtered \* Math.log(filtered) \*\* 2)}`;

});

Файл index.html:

<!DOCTYPE html>

<html>

<head>

<link rel="stylesheet" href="index.css" />

</head>

<body>

<div>

<form id="form" class="form">

<textarea oninput="hanldeChange(this.value)"></textarea>

<button>Show</button>

</form>

<p id="words\_count">Кол-во слов</p>

<p id="in\_out">Ген. совок-ть вход/выход</p>

<p id="v">V =</p>

<p id="v\_">V\* =</p>

<p id="l">L =</p>

</div>

<script src="index.js"></script>

</body>

</html>

Файл index.css:

.form {

width: 80%;

min-width: 500px;

height: 400px;

}

.form textarea {

width: 100%;

height: 80%;

font-size: 12px;

}